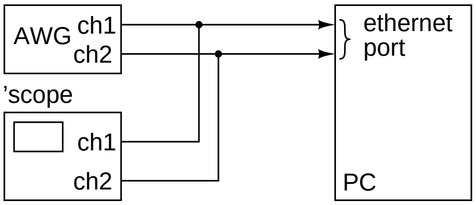
ELEX 3525 : Data Communications
2020 Fall Term

# 10 Mb/s Ethernet Transmitter

## Introduction

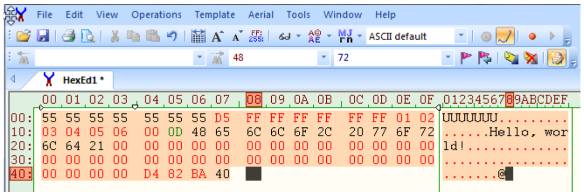
In this lab you will use a hex editor to create a file containing an Ethernet frame. Then you will use a supplied program to convert this file to a 10 Mb/s Ethernet waveform in `.csv` format. You will load this file into the AWG which will generate an Ethernet waveform that will be received by the Ethernet interface on the lab PC. You will use a 'scope to measure the interface voltages and signal timing. You will use the Wireshark protocol analyzer to display the frame contents.



## Create an Ethernet Frame

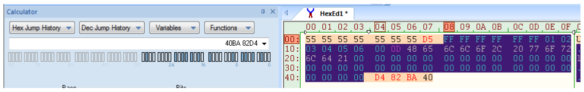
Run HexEdit (available on the course web site) and use the menu item File⇒New to create a new file. Select a file size of 0 bytes. Right-click in the data area and select "Allow Changes". Use the Insert key to switch between inserting and over-writing existing values. Use the Tab key to switch between entering values as hex or ASCII characters.

Enter the bytes of an Ethernet packet that contains your full name and student ID as the payload (data). Include the preamble and SFD. The destination address should be the broadcast address (all 0xff). The source address should be the last six digits of your student ID (e.g. if you ID was A00123456 you would use a source address of `01:02:03:04:05:06`). Set the length/type field to the number of bytes in the payload (i.e. in your name). This value will prevent decoding of the payload because it will not correspond to a known Ethertype (e.g. IP). The data field should contain your name padded with zeros to reach the minimum frame length.



```
     00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F  0123456789ABCDEF
00:  55 55 55 55  55 55 55 D5  FF FF FF FF  FF FF 01 02  UUUUUUU.........
10:  03 04 05 06  00 0D 48 65  6C 6C 6F 2C  20 77 6F 72  .......Hello, wor
20:  6C 64 21 00  00 00 00 00  00 00 00 00  00 00 00 00  ld!.............
30:  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
40:  00 00 00 00  D4 82 BA 40                            .......@
```

To compute the FCS, select the bytes *following the preamble* and compute the 32-bit CRC using the menu item Operations⇒Checksum⇒CRC 32. Add this 4-byte value to the end of the packet. Note that the CRC value computed by HexEdit is a 32-bit value that must be transmitted in order from LS to MS bit (you will have to reverse the byte order and the program will reverse the bit order).



```
40BA 82D4
```

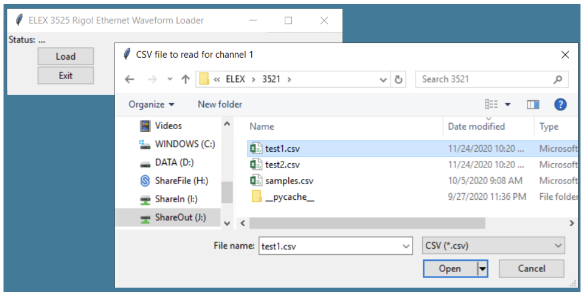
```
     00 01 02 03  04 05 06 07  08 09 0A 0B  0C 0D 0E 0F
00:  55 55 55 55  55 55 55 D5  FF FF FF FF  FF FF 01 02
10:  03 04 05 06  00 0D 48 65  6C 6C 6F 2C  20 77 6F 72
20:  6C 64 21 00  00 00 00 00  00 00 00 00  00 00 00 00
30:  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
40:  00 00 00 00  D4 82 BA 40
```

Use File⇒Save to save the data to a file called `frame.bin`. Make sure the length of the file in bytes is the minimum Ethernet frame length[1] plus the preamble length (including the SFD).

Create the `frame.bin` file according to the instructions above.

## Configure the AWG and Verify Waveform

Use the `etherload.py` script in the ELEX 3521 folder on ShareOut to load the `test1.csv` and `test2.csv` files (in the same folder) into the AWG:



Status: ...

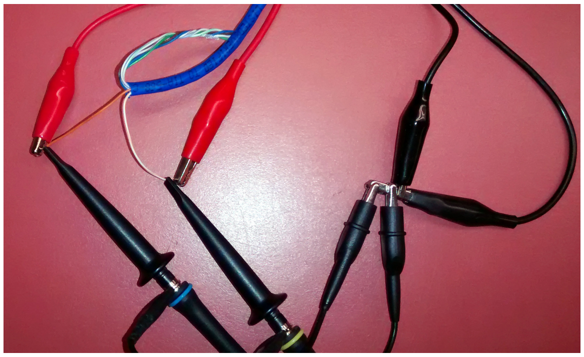
| Name | Date modified | Type |
|---|---|---|
| test1.csv | 11/24/2020 10:20 ... | Microsoft |
| test2.csv | 11/24/2020 10:20 ... | Microsoft |
| samples.csv | 10/5/2020 9:08 AM | Microsoft |
| __pycache__ | 9/27/2020 11:36 PM | File folder |

File name: test1.csv
CSV (*.csv)

[1] Unless your name is very long, it should fit into the shortest-allowable Ethernet frame.

The configuration of the AWG is somewhat complicated by the need to use both channels to obtain a differential output. The AWG also has be set up in burst mode so that one waveform (frame) is transmitted every 16ms. This is the time interval between the NLP ("normal link") pulses that are included in the waveform created by the software.

In the lab, the two AWG outputs have been connected to the two 'scope channels and to a cable connected to the Ethernet adapter in the lab PC as shown below:

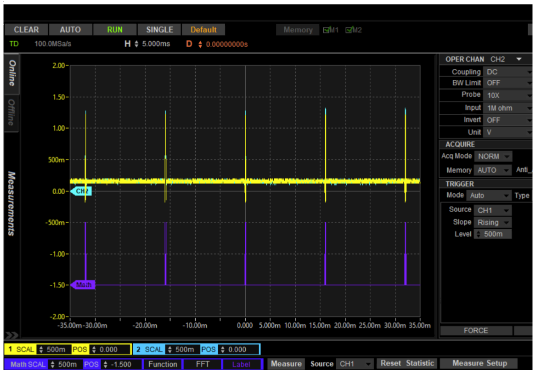


Use the UltraScope remote oscilloscope interface to check that the voltage levels, signal period and the preamble waveform are as expected.

Configure Ultrascope with the settings shown below. Note that the signal is connected to the 'scope through 10X probes. The purple trace is the "Math" trace which is configured using the Ultrascope "Function" button. Set the function to A-B with A being channel 1 and B channel 2.

Ultrascope, has a very limited set of triggering controls compared to the ones available through the front panel, but you should be able to configure it to see the following:

- The Ethernet frames being repeated every 16ms:



TD 100.0MSa/s H 5.000ms D 0.00000000s

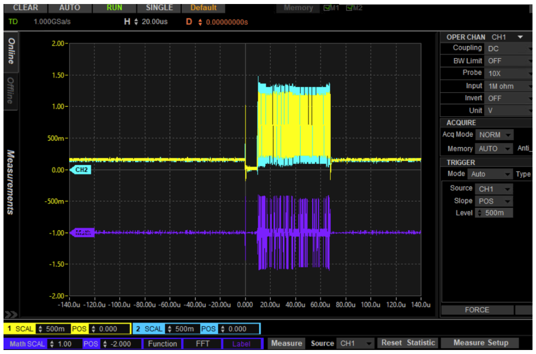
OPER CHAN CH2<br>Coupling DC<br>BW Limit OFF<br>Probe 10X<br>Input 1M ohm<br>Invert OFF<br>Unit V

ACQUIRE<br>Acq Mode NORM<br>Memory AUTO

TRIGGER<br>Mode Auto<br>Source CH1<br>Slope Rising<br>Level 500m

1 SCAL 500m POS 0.000 2 SCAL 500m POS 0.000 Math SCAL 500m POS -1.500

- the NLP pulse followed by the Ethernet frame:



TD 1.000GSa/s H 20.00us D 0.00000000s

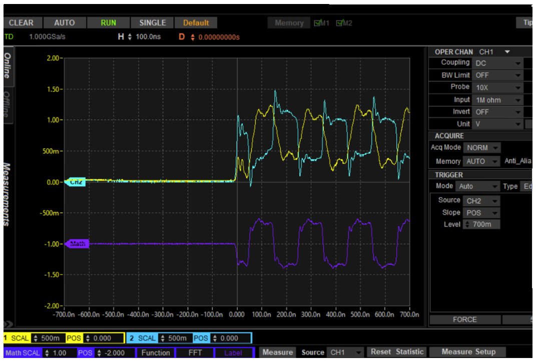
OPER CHAN CH1<br>Coupling DC<br>BW Limit OFF<br>Probe 10X<br>Input 1M ohm<br>Invert OFF<br>Unit V

ACQUIRE<br>Acq Mode NORM<br>Memory AUTO

TRIGGER<br>Mode Auto<br>Source CH1<br>Slope POS<br>Level 500m

1 SCAL 500m POS 0.000 2 SCAL 500m POS 0.000 Math SCAL 1.00 POS -2.000

- the first few bits of the frame showing the two channels and how the differential signal, computed using Math mode, is a Manchester-coded signal with period of 100 ns per bit and levels of ±1. You will need to trigger using channel 2 and increase the trigger level to trigger on the first data bit instead of the NLP pulse.



TD 1.000GSa/s H 100.0ns D 0.00000000s

OPER CHAN CH1<br>Coupling DC<br>BW Limit OFF<br>Probe 10X<br>Input 1M ohm<br>Invert OFF<br>Unit V

ACQUIRE<br>Acq Mode NORM<br>Memory AUTO

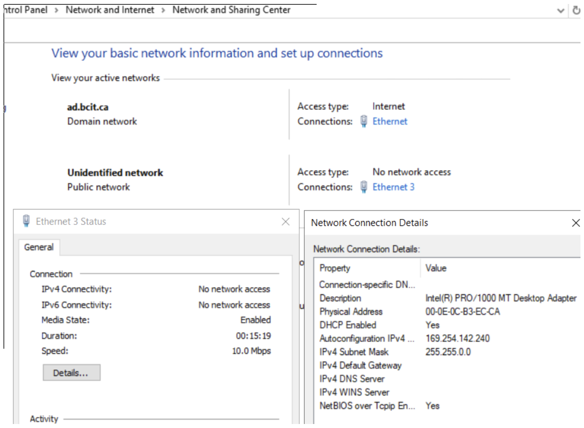
TRIGGER<br>Mode Auto<br>Source CH2<br>Slope POS<br>Level 700m

1 SCAL 500m POS 0.000 2 SCAL 500m POS 0.000 Math SCAL 1.00 POS -2.000

Obtain a screenshot of the start of the frame for use in your report.

## View Packet Contents Using Wireshark

From the Start menu search for "View Network Status". Find the network interface with the Intel network interface card (NIC):


ntrol Panel › Network and Internet › Network and Sharing Center
Details...

View your basic network information and set up connections

View your active networks

**ad.bcit.ca**
Domain network

Access type: Internet
Connections: Ethernet

**Unidentified network**
Public network

Access type: No network access
Connections: Ethernet 3

Ethernet 3 Status

General

Connection

| | |
|---|---|
| IPv4 Connectivity: | No network access |
| IPv6 Connectivity: | No network access |
| Media State: | Enabled |
| Duration: | 00:15:19 |
| Speed: | 10.0 Mbps |

Activity

Network Connection Details

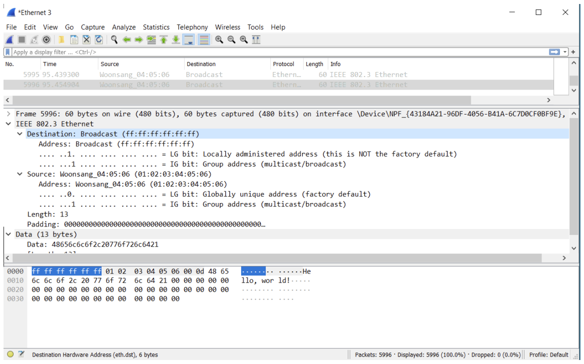
Network Connection Details:

| Property | Value |
|---|---|
| Connection-specific DN... | |
| Description | Intel(R) PRO/1000 MT Desktop Adapter |
| Physical Address | 00-0E-0C-B3-EC-CA |
| DHCP Enabled | Yes |
| Autoconfiguration IPv4 ... | 169.254.142.240 |
| IPv4 Subnet Mask | 255.255.0.0 |
| IPv4 Default Gateway | |
| IPv4 DNS Server | |
| IPv4 WINS Server | |
| NetBIOS over Tcpip En... | Yes |

Run the Wireshark protocol analyzer, and double-click on the network interface found above to start capturing frames.

If the Type/Length field is less than 0x600 Wireshark will treat the packet as IEEE 802.3. The rarely-used 802.3 format uses the Type/Length field as a length field and includes a 4-byte LLC header following the type length field. Disable decoding of the 802.2 LLC protocol under the menu item Analyzer / Enabled Protocols... and de-selecting the LLC checkbox.

You should now see packets being displayed that say “Hello world” in the data portion as shown in the screen capture below.


*Ethernet 3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time        Source              Destination  Protocol  Length Info
 5995 95.439300   Woonsang_04:05:06   Broadcast    Ethern...     60 IEEE 802.3 Ethernet
 5996 95.454904   Woonsang_04:05:06   Broadcast    Ethern...     60 IEEE 802.3 Ethernet
```

```
> Frame 5996: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{43184A21-96DF-4056-B41A-6C7D0CF0BF9E},
v IEEE 802.3 Ethernet
  v Destination: Broadcast (ff:ff:ff:ff:ff:ff)
      Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
  v Source: Woonsang_04:05:06 (01:02:03:04:05:06)
      Address: Woonsang_04:05:06 (01:02:03:04:05:06)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Length: 13
    Padding: 0000000000000000000000000000000000000000000…
v Data (13 bytes)
    Data: 48656c6c6f2c20776f726c6421
```

```
0000  ff ff ff ff ff ff 01 02  03 04 05 06 00 0d 48 65   ........ ......He
0010  6c 6c 6f 2c 20 77 6f 72  6c 64 21 00 00 00 00 00   llo, wor ld!.....
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

Destination Hardware Address (eth.dst), 6 bytes    Packets: 5996 · Displayed: 5996 (100.0%) · Dropped: 0 (0.0%)    Profile: Default

## Convert Frame Data to Waveform

Now that you have verified that everything is configured and connected properly, you need to transmit the Ethernet frame you prepared earlier to verify it's correct.

The `etherlab.c` program reads a file and encodes the bits as a Manchester-encoded waveform. The bits are encoded in order from LS to MS bit. Therefore, if the bytes in your file represent a complete and correct Ethernet frame (including preamble and CRC) the output waveforms will represent a valid 10BASE-T frame.

The program reads a file called **`frame.bin`** and generates two files, **`channel1.csv`** and **`channel2.csv`** corresponding to the files that should be output on channels 1 and 2 of the AWG respectively.

Copy the compiled program, **`etherlab.exe`**, to the folder in which you've put your **`frame.bin`** file. Run the program (Shift-right-click, select “Open Powershell Here” and enter **`.\etherlab.exe`**). The program should create the **`channel`*n*`.csv`** files. Copy these to a folder on the lab PC.

## Check your Frame

Run the **`etherload.py`** script again and select your **`channel1.csv`** file to load the two **`.csv`** files into the AWG as described earlier and view the decoded frames using Wireshark.

Take a screen capture (or two) showing the contents of the 802.3 header and the complete contents of the packet in both hex and ASCII (as shown above) for your report.

## Report

Submit a report including the following:

- the usual identification information
- a printout from HexEdit showing your packet's contents in hex and ASCII
- a 'scope screen capture showing the single-ended and differential voltages at the start of the packet showing the start of the preamble
- screen captures from Wireshark showing the received Ethernet header and data fields.
- Answers to the following questions:

  (1) What is the minimum length of an 802.3 frame, not including the preamble? Does this include the header and FCS?

(2) What is the frequency of the preamble waveform[2]? Why?

(3) Assuming the minimum frame size, how many bytes will be transmitted per frame, including preamble and FCS?

(4) What part(s) of the frame you created does Wireshark not show?

[2] Read the square wave's period from the 'scope screen capture and compute the frequency. The answer may not be what you expect.